

Appl. No. 09/784,392
Amndt. dated June 2, 2004
Reply to Office of April 5, 2004

REMARKS

In the Office Action dated April 5, 2004, claims 1-9 and 17-43 were rejected under 35 U.S.C. § 103 over U.S. Patent No. 5,544,359 (Tada) in view of U.S. Patent No. 5,864,849 (Bohannon); and claims 10-16 were rejected under § 103 over Tada alone.

It is respectfully noted that Tada does not disclose the flushing of a transaction log from a volatile storage to non-volatile storage by each access module before an end transaction procedure. As recited in claim 1, flushing of the transaction log to non-volatile storage occurs *before* an end transaction procedure. This clearly is not disclosed by Tada, contrary to the assertion made in the Office Action. Because the Office Action has mis-applied Tada against an element of claim 1, the obviousness rejection is defective on at least this ground.

The Office Action cited to Figure 5 and the passage at column 10, lines 33-34, of Tada as disclosing the performing of a flush of a transaction log. *See* 4/5/2004 Office Action at 4. The cited passage refers to the transfer of log data to a log data buffer 132 on a main storage unit 101. Tada, 10:33-34. The main storage unit "is formed of a *volatile* random-access memory (RAM)." Tada, 8:13-14 (emphasis added). The log data buffer 132 (in addition to volatile HLF buffers 114) are part of the volatile main storage unit 101. Tada, 8:5-9. Thus, the transfer of the log data to log data buffer 132 is a transfer of data to *volatile* storage. Therefore, the cited passage in column 10 of Tada does not disclose the performance of flushing of a transaction log from volatile storage to non-volatile storage.

In Tada, the transfer of log data to HLF buffer 112 in non-volatile memory 103 does not occur until *after* the issuance of a transaction-end instruction. Tada, 10:43-44, 11:30-33. Therefore, the transfer of log data from volatile storage to non-volatile storage as performed in Tada is *after* (not before) a transaction-end instruction has occurred--in other words, the transfer of log data from volatile storage to non-volatile storage is *part* of an end transaction procedure, not *before* an end transaction procedure, as recited in claim 1.

The present Office Action further stated that "it would have been obvious to one of ordinary skill in the art at the time the invention was made to flush the log to non-

Appl. No. 09/784,392
Amdt. dated June 2, 2004
Reply to Office of April 5, 2004

volatile storage so that the data will not be lost when the system goes down." 4/5/2004 Office Action at 2, 4. The present Office Action also referred to Bohannon as teaching the flushing of the transaction log to non-volatile storage. 4/5/2004 Office Action at 2. While it may be known to flush a transaction log to non-volatile storage, there is no teaching or suggestion anywhere within any of the cited references of flushing the transaction log to non-volatile storage *before* an end transaction procedure. In fact, the opposite is suggested by Tada, which teaches writing of log data to non-volatile historical log files (HLF) in step S09 in Figure 5 of Tada. Step S09 is performed after the system has issued a transaction-end macro (step S06 in Figure 5 of Tada). *See* Tada, 10:43-44. In other words, in Tada, the writing of a transaction log from volatile storage to non-volatile storage occurs during an end transaction procedure, not before an end transaction procedure. Performing a flush of a transaction log to non-volatile storage before an end transaction procedure enables more efficient processing, as discussed in the present application. One of the benefits achieved by the present application is, in certain cases, flushing of a transaction log can be *avoided* during an end transaction procedure. *See* Specification, page 13, lines 4-10. As a result, end transaction processing is made more efficient. The ability to perform a flush of a transaction log from volatile storage to non-volatile storage before an end transaction procedure is clearly not even remotely suggested by Tada.

Bohannon similarly teaches that the flushing of volatile tail 175 to non-volatile memory occurs during transaction *commit*, which is part of the end transaction procedure. *See* Bohannon, 10:4-9. Therefore, Bohannon is similar to Tada in teaching that the flushing of a transaction log occurs during, not before, an end transaction procedure. In view of the foregoing, even if Tada and Bohannon can be properly combined, the combination does not teach or suggest each and every element of claim 1. Therefore, a *prima facie* case of obviousness has not been established with respect to claim 1.

Furthermore, there simply is no motivation to modify the teachings of Tada to achieve the claimed invention of flushing a transaction log to non-volatile storage before an end transaction procedure. Both Tada and Bohannon suggest the complete opposite, namely, that flushing the transaction log is performed during an end transaction procedure. Thus, a person of ordinary skill in the art at the time of the invention, looking

Appl. No. 09/784,392
Amdt. dated June 2, 2004
Reply to Office of April 5, 2004

to the teachings of Tada and Bohannon, would have been led to performing transaction log flushing to non-volatile storage during end transaction processing, not before end transaction processing. Therefore, such a person of ordinary skill in the art would not have been motivated to achieve the claimed invention.

Therefore, because of a lack of motivation or suggestion to combine or modify the teachings of Tada and Bohannon to achieve the claimed invention, the *prima facie* case of obviousness is defective for this further reason.

Withdrawal of the obviousness rejection of claim 1 is therefore respectfully requested.

Independent claims 17, 21, 24, and 28 are similarly allowable over the asserted combination of Tada and Bohannon. In fact, independent claim 28 even expressly recites that the parsing engine is adapted to avoid sending a broadcast directive to access modules to cause performance of a transaction log flush during the end transaction procedure. Withdrawal of the § 103 rejections of claims 17, 21, 24, and 28 is respectfully requested.

Independent claim 10 was rejected as being obvious over Tada alone. The Office Action conceded that Tada does not disclose a fallback module. However, without citing to any actual reference that would suggest a modification of Tada to achieve the subject matter of claim 10, the Office Action stated that the invention of claim 10 would be obvious over Tada.

Claim 10 recites a first access module in a database system writing an end transaction indication to a first transaction log portion, where the first access module is *part of a cluster of access modules*. Claim 10 also recites the first access module sending an end transaction directive to a *fallback module* associated with the first access module, where the fallback module is also part of the cluster. The Office Action stated that Tada somehow suggests the acts performed in claim 10, even though the Office Action conceded, with respect to claim 1, that "Tada does not explicitly disclose the transaction as processed by a plurality of access modules." 4/5/2004 Office Action at 4. In fact, in the rejection of claim 10, the Office Action made no mention whatsoever of the fact that claim 10 recites that the first access module is part of a cluster of *access modules* (note

Appl. No. 09/784,392
Amdt. dated June 2, 2004
Reply to Office of April 5, 2004

the plural sense of "access modules"). On at least this basis alone, the rejection of claim 10 is defective.

The Office Action then stated that "[f]allback is a redundancy operation in which a copy of a database portion is stored on a different access module than where the original data portion is stored. Tada teaches transferring data to buffer which a different storage location than where the original of the data portion is stored (Tada, col. 10, lines 29-67)." 4/5/2004 Office Action at 14-15. Although the cited column 10 passage of Tada describes issuing a transaction-end macro instruction, the transfer of log data to a log data buffer 132, and the transfer of the extracted log data to HLF buffer 114, there is absolutely no indication whatsoever of a first access module being part of a cluster of access modules, or a first access module sending an end transaction directive to a fallback module, where the fallback module is also part of the cluster. The Office Action further stated that "[i]t is unclear to the Examiner what is a fallback module; therefore the Examiner interprets a fallback module as any processing module in a computer system." 4/5/2004 Office Action at 2. Applicant respectfully submits that the term "fallback" is well defined in the specification. *See* Specification, page 15, line 17-page 18, line 7. The role of a "fallback access module" in accordance with an embodiment is also described on pages 15-18 of the Specification. Thus, the term "fallback module" is well supported in the Specification. Therefore, the term "fallback module" cannot be arbitrarily construed to mean any processing module. In view of the foregoing, the obviousness rejection of claim 10 is defective.

All dependent claims are allowable for at least the same reasons as corresponding independent claims. Allowance of all claims is respectfully requested. The Commissioner is authorized to charge any additional fees and/or credit any overpayment to Deposit Account No. 50-1673 (9417).

Appl. No. 09/784,392
Amdt. dated June 2, 2004
Reply to Office of April 5, 2004

Respectfully submitted,



June 2, 2004
Date

Dan C. Hu, Reg. No. 40,025
Trop, Pruner & Hu, P.C.
8554 Katy Freeway, Ste. 100
Houston, TX 77024
713/468-8880
713/468-8883 [fax]